RSAC | 2025 Conference

Many Voices.
One Community.

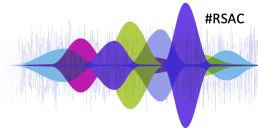
SESSION ID: CRYP-T01A

Simplified Adaptively Secure Threshold BLS Signatures

Benoît Libert

Zama, France

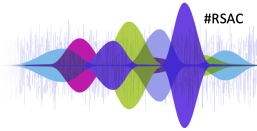
Outline



1. Threshold Cryptography
   - Motivation
   - Static vs Adaptive corruptions
   - Security Definition

2. Adaptively Secure Threshold BLS Signatures
   - The Das-Ren Scheme and its proof intuition
   - Our Simplified Scheme
   - Comparison

Threshold Cryptography



- Splitting secret keys among shareholders (`Desmedt-Frankel, Crypto'89; Boyd, IMA'89`):

  Enhances the security of highly sensitive keys and the availability of systems

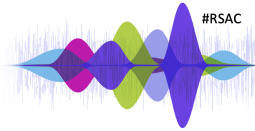
- **Motivation:** Avoid single points of failure

  - Attacks on certification authorities lead to fake certificates distributed over the Internet and destroy the "trust infrastructure"

    e.g., DigiNotar was hacked and attacker produced a DigiNotar-signed "Google certificate" (September 2011)

  - Also useful in blockchain applications

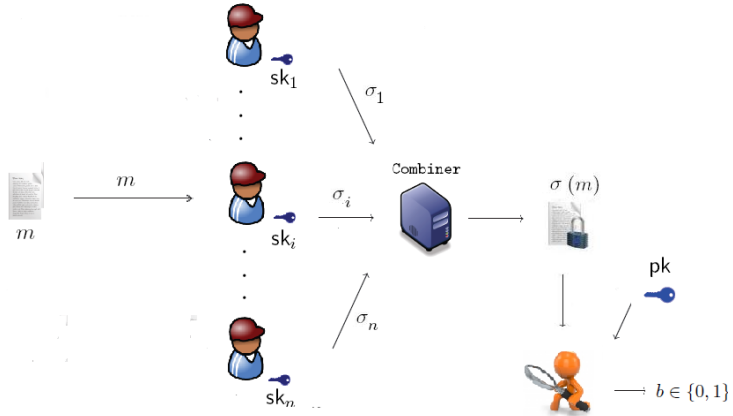
Threshold Cryptography

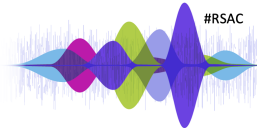


- Split private keys into n shares $SK_1, \ldots, SK_n$ so that knowing strictly less than $t + 1 < n$ shares is useless to the adversary
- Quorum of $t + 1 < n$ shareholders must contribute to private key operations
- Key generation is either centralized (trusted dealer) or distributed
- Useful properties:
  - Robustness: up to $t < n$ malicious servers cannot prevent an honest majority from decrypting/signing
  - Optimal resilience: robustness against any minority $t \leq (n - 1)/2$
  - Non-interactivity: no communication among shareholders
  - No reliance on reliable erasures

Threshold Cryptography

Non-interactive (t, n)-threshold signatures

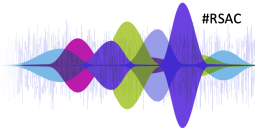




Static Corruptions

Adversary corrupts servers *before* seeing the public key

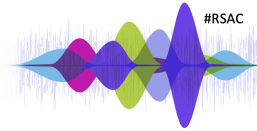
- First threshold signatures (non-exhaustive):
  - `Desmedt-Frankel (Crypto'91)`: threshold RSA w/o robustness
  - `De Santis` *et al.* `(STOC'94)`: provably secure, but large partial signatures
  - `Gennaro` *et al.* `(Eurocrypt'96 & Crypto'96)`: threshold DSA & RSA signatures



Static Corruptions

Adversary corrupts servers *before* seeing the public key

- First threshold signatures (non-exhaustive):
  - `Desmedt-Frankel (Crypto'91)`: threshold RSA w/o robustness
  - `De Santis` *et al.* `(STOC'94)`: provably secure, but large partial signatures
  - `Gennaro` *et al.* `(Eurocrypt'96 & Crypto'96)`: threshold DSA & RSA signatures
- Robust threshold signatures without interaction:
  - `Shoup (Eurocrypt'00)`: threshold RSA signatures
  - `Katz-Yung (Asiacrypt'02)`: threshold Rabin signatures
  - `Boldyreva (PKC'03)`: short threshold BLS signatures
  - `Boneh` *et al.* `(Crypto'18)`: universal thresholdizers



Adaptive Corruptions

Adversary corrupts up to t servers *at any time*

- With interaction (non-exhaustive)
  - Assuming erasures
    (Canetti *et al.*, Crypto'99; Frankel-MacKenzie-Yung, ESA'99, Asiacrypt'99)
  - Without erasures (Jarecki-Lysyanskaya, Eurocrypt'00; Abe-Fehr, Crypto'04)
  - Optimal resilience, better efficiency (Bacho *et al.*, Eurocrypt'24)
- Without interaction
  - Short signatures (2 group elements) without interaction, erasures, or trusted dealer
    (Libert-Joye-Yung, PODC'14)
  - Adaptive security in the AGM for threshold BLS (Bacho-Loss, CCS'22)
  - Optimally short threshold BLS in the ROM (Das-Ren, Crypto'24)

Our Contribution

- A simpler version of the Das-Ren threshold BLS signature:
  - Shorter secret key shares (2 elements of $\mathbb{Z}_p$ instead of 3)
  - Shorter partial signatures (by $\approx 20\%$)
  - Faster partial signing (by $\approx 25\%$) and verification (by $\approx 33\%$)
  - Smaller verification keys (by $\approx 50\%$)
  - Smaller communication in distributed key generation (by $\approx 50\%$)
- …while retaining advantages
  - Adaptive security without erasures; no interaction among signers
  - Optimal-size signatures
  - One-round DKG

Non-interactive Threshold Signatures

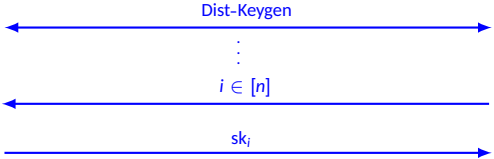
Security under chosen-message attacks:

Dist-Keygen

$\{sk_i\}_{i \in C}$, pk

Non-interactive Threshold Signatures

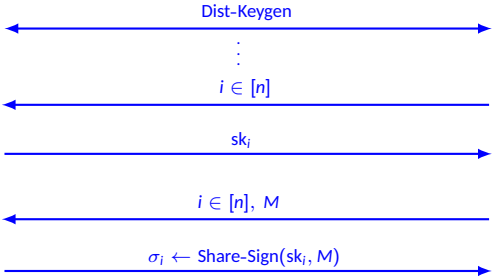
Security under chosen-message attacks:



$\{sk_i\}_{i \in C}$, pk

Non-interactive Threshold Signatures

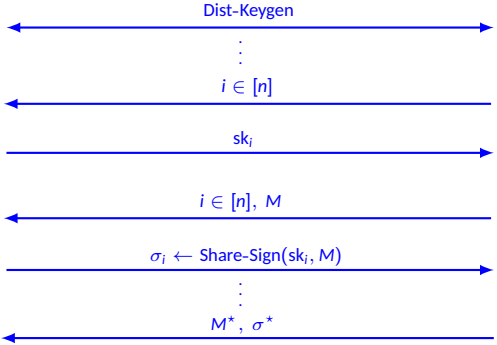
Security under chosen-message attacks:



$\{\mathsf{sk}_i\}_{i \in \mathcal{C}},\ \mathsf{pk}$

Non-interactive Threshold Signatures

Security under chosen-message attacks:

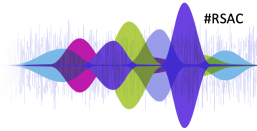


$\{\mathsf{sk}_i\}_{i \in \mathcal{C}},\ \mathsf{pk}$

Adversary wins if:

- $\text{Verify}(\mathsf{pk}, M^\star, \sigma^\star) = 1$
- $|\mathcal{V} \cup \mathcal{C}| \leq t$ where $\mathcal{V} := \{i \in \{1, \ldots, n\} \mid (i, M^\star) \text{ was queried}\}$

BLS Signatures (Boneh-Lynn-Shacham, Asiacrypt'01)



- Based on bilinear maps $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ s.t.

$$e(g^a, \hat{h}^b) = e(g^b, \hat{h}^a) = e(g, \hat{h})^{ab} \qquad \forall g \in \mathbb{G},\ \hat{h} \in \hat{\mathbb{G}},\ a, b \in \mathbb{Z}$$

- Uses a random oracle $H : \{0,1\}^* \to \mathbb{G}$; Public key is $\hat{g}^s$
- Signature is

$$\sigma = H(M)^s \in \mathbb{G}$$

- Verification checks if $e(\sigma, \hat{g}) = e(H(M), \hat{g}^s)$
- Easy to thresholdize (Boldyreva, PKC'03): optimal-size secret-key shares, static security

Hardness Assumptions

The **Decision Diffie-Hellman (DDH)** problem

In pairing-frienly groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p: distinguish between

$$\left\{ \left(g,\ g^a,\ g^b,\ g^{ab}\right) \in \mathbb{G}^4 \mid a, b \xleftarrow{R} \mathbb{Z}_p \right\}, \quad \left\{ \left(g,\ g^a,\ g^b,\ g^c\right) \in \mathbb{G}^4 \mid a, b, c \xleftarrow{R} \mathbb{Z}_p \right\}$$

The **co-Diffie-Hellman (co-CDH)** problem

In pairing-frienly groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$: given

$$\left(g,\ g^a,\ g^b\right) \in \mathbb{G}^3, \qquad \left(\hat{g},\ \hat{g}^a\right) \in \hat{\mathbb{G}}^2$$

with $a, b \xleftarrow{R} \mathbb{Z}_p$, find $g^{ab} \in \mathbb{G}$

The Das-Ren Scheme

- Public key is

$$\mathsf{pk} = \left(\hat{g},\ \hat{h},\ \hat{v},\ \hat{g}^{s} \right) \in \hat{\mathbb{G}}^4$$

- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i), u(i)) \}_{i \in [n]}$ via

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t, \qquad u[X] = 0 + u_1 X + \cdots + u_t X^t$$

Verification keys are $\left\{ \mathsf{vk}_i = \hat{g}^{s(i)} \cdot \hat{h}^{r(i)} \cdot \hat{v}^{u(i)} \in \hat{\mathbb{G}} \right\}_{i \in [n]}$

The Das-Ren Scheme

- Public key is

$$\mathsf{pk} = \left(\hat{g},\ \hat{h},\ \hat{v},\ \hat{g}^{s} \right) \in \hat{\mathbb{G}}^4$$

- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i), u(i)) \}_{i \in [n]}$ via

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t, \qquad u[X] = 0 + u_1 X + \cdots + u_t X^t$$

  Verification keys are $\left\{ \mathsf{vk}_i = \hat{g}^{s(i)} \cdot \hat{h}^{r(i)} \cdot \hat{v}^{u(i)} \in \hat{\mathbb{G}} \right\}_{i \in [n]}$

- Partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ are proven consistent with vk_i

The Das-Ren Scheme

- Public key is

$$\mathsf{pk} = \left(\hat{g},\ \hat{h},\ \hat{v},\ \hat{g}^{s} \right) \in \hat{\mathbb{G}}^4$$

- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i), u(i)) \}_{i \in [n]}$ via

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t, \qquad u[X] = 0 + u_1 X + \cdots + u_t X^t$$

  Verification keys are $\left\{ \mathsf{vk}_i = \hat{g}^{s(i)} \cdot \hat{h}^{r(i)} \cdot \hat{v}^{u(i)} \in \hat{\mathbb{G}} \right\}_{i \in [n]}$

- Partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ are proven consistent with vk_i
- For any $(t+1)$-set $S \subset [n]$,

$$\sigma = \prod_{i \in S} \sigma_i^{\Delta_{i,S}(0)} = H_0(M)^s$$

Adaptive Security: Intuition

Theorem (Das-Ren, Crypto'24)

In the ROM, the fully distributed scheme is adaptively secure for any $t \leq (n-1)/2$ *under the co-CDH assumption in* $(\mathbb{G}, \hat{\mathbb{G}})$ *and the DDH assumption in* $\mathbb{G}$

Proof intuition:

- Make partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ information-theoretically hide $s(i)$

  Move to a game where $\{(H_0(M), H_1(M))\}_{M \neq M^\star}$ live in a 1-dimension subspace (using DDH in $\mathbb{G}$)

Adaptive Security: Intuition

Theorem (Das-Ren, Crypto'24)

In the ROM, the fully distributed scheme is adaptively secure for any $t \leq (n-1)/2$ *under the co-CDH assumption in* $(\mathbb{G}, \hat{\mathbb{G}})$ *and the DDH assumption in* $\mathbb{G}$

Proof intuition:

- Make partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ information-theoretically hide $s(i)$

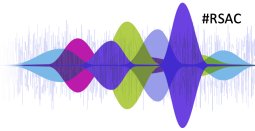
  Move to a game where $\{(H_0(M), H_1(M))\}_{M \neq M^\star}$ live in a 1-dimension subspace (using DDH in $\mathbb{G}$)

- Switch to $r[X]$ s.t. $r(0) = 1$ for a "rigged"

$$\mathsf{pk} = \hat{g}^{s(0)} \cdot \hat{h}^{r(0)} = \hat{g}^{s(0)+a}$$

  Forgery $\sigma^\star = H_0(M^\star)^{s(0)+a}$ reveals co-CDH solution g^{ab} if $H_0(M^\star) = g^b$

Adaptive Security: Intuition



Theorem (Das-Ren, Crypto'24)

In the ROM, the fully distributed scheme is adaptively secure for any $t \leq (n-1)/2$ *under the co-CDH assumption in* $(\mathbb{G}, \hat{\mathbb{G}})$ *and the DDH assumption in* $\mathbb{G}$

Proof intuition (cont.):

- **Difficulty:** Hide the replacement of $r(0) = 0$ by $r(0) = 1$
- Das-Ren uses $\mathsf{vk}_i = \hat{g}^{s(i)} \cdot \hat{h}^{r(i)} \cdot \hat{v}^{u(i)}$ that statistically hides $(s(i), r(i))$

Adaptive Security: Intuition

Theorem (Das-Ren, Crypto'24)

In the ROM, the fully distributed scheme is adaptively secure for any $t \leq (n-1)/2$ *under the co-CDH assumption in* $(\mathbb{G}, \hat{\mathbb{G}})$ *and the DDH assumption in* $\mathbb{G}$

Proof intuition (cont.):

- **Difficulty:** Hide the replacement of $r(0) = 0$ by $r(0) = 1$
- Das-Ren uses $\mathsf{vk}_i = \hat{g}^{s(i)} \cdot \hat{h}^{r(i)} \cdot \hat{v}^{u(i)}$ that statistically hides $(s(i), r(i))$
- **Our solution**: Make the information revealed by vk_i redundant with $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$
  - Verification keys are $\mathsf{vk}_i = g^{s(i)} \cdot h^{r(i)}$ (live in $\mathbb{G}$ instead of $\hat{\mathbb{G}}$)
  - Make $\{(H_0(M), H_1(M))\}_{M \neq M^\star}$ colinear with $(g, h) \in \mathbb{G}^2$ using DDH

Our Simplified Scheme

- Public key is

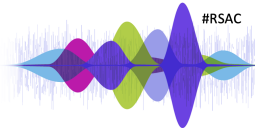
$$\mathsf{pk} = \left((g,\ h) \in \mathbb{G}^2,\ (\hat{g},\ \hat{g}^s) \in \hat{\mathbb{G}}^2 \right)$$

- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i)) \}_{i \in [n]}$ using

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t$$

Verification keys are $\left\{ \mathsf{vk}_i = g^{s(i)} \cdot h^{r(i)} \in \mathbb{G} \right\}_{i \in [n]}$

Our Simplified Scheme



- Public key is

$$\mathsf{pk} = \left((g,\ h) \in \mathbb{G}^2,\ (\hat{g},\ \hat{g}^s) \in \hat{\mathbb{G}}^2 \right)$$

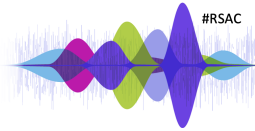
- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i)) \}_{i \in [n]}$ using

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t$$

  Verification keys are $\left\{ \mathsf{vk}_i = g^{s(i)} \cdot h^{r(i)} \in \mathbb{G} \right\}_{i \in [n]}$

- Partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ are proven consistent with vk_i

Our Simplified Scheme



- Public key is

$$\mathsf{pk} = \left((g,\ h) \in \mathbb{G}^2,\ (\hat{g},\ \hat{g}^s) \in \hat{\mathbb{G}}^2 \right)$$

- $s \in \mathbb{Z}_p$ is shared as $\{ \mathsf{sk}_i = (s(i), r(i)) \}_{i \in [n]}$ using

$$s[X] = s + s_1 X + \cdots + s_t X^t, \qquad r[X] = 0 + r_1 X + \cdots + r_t X^t$$

  Verification keys are $\left\{ \mathsf{vk}_i = g^{s(i)} \cdot h^{r(i)} \in \mathbb{G} \right\}_{i \in [n]}$

- Partial signatures $\sigma_i = H_0(M)^{s(i)} \cdot H_1(M)^{r(i)}$ are proven consistent with vk_i

- For any authorized set $S \subset [n]$, we get $\sigma = \prod_{i \in S} \sigma_i^{\Delta_{i,S}(0)} = H_0(M)^s$

Security

Theorem

In the ROM, the fully distributed scheme is adaptively secure for any $t \leq (n-1)/2$ under the co-CDH assumption in $(\mathbb{G}, \hat{\mathbb{G}})$ and the DDH assumption in $\mathbb{G}$

Proof idea:

- (Modified) DKG protocol only uses $g, h = g^a \in \mathbb{G}$; $\hat{g}^a$ is *not* public
- Choose a random $i^\star \in [n]$ that won't be corrupted (with probability $> 1/2$)
  (`Frankel-MacKenzie-Yung, ESA'99`; `Canetti et al., Crypto'99`)
- Use DDH to make $\{(H_0(M), H_1(M))\}_{M \neq M^\star}$ colinear with $(g, h) \in \mathbb{G}^2$
- "Program" $r[X]$ to have $r(0) = 1$ by making player $i^\star \in [n]$ share $r_{i^\star}(0) = 1$ in the DKG
- Prove that an adversary corrupting $\leq t$ players cannot detect the switch

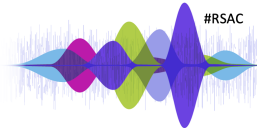
Comparison

	Assumptions	$\lvert\{\mathsf{vk}_i\}_{i=1}^n\rvert$	$\lvert\mathsf{sk}_i\rvert$	$\lvert\sigma_i\rvert$	Partial signing cost	Partial verification cost
Boldyreva*	co-CDH	$n \times \lvert\hat{\mathbb{G}}\rvert$	$1 \times \lvert\mathbb{Z}_p\rvert$	$1 \times \lvert\mathbb{G}\rvert$	1 $\mathsf{exp}_{\mathbb{G}}$	2 pairings
Das-Ren	co-CDH + $\mathsf{DDH}_{\mathbb{G}}$	$n \times \lvert\hat{\mathbb{G}}\rvert$	$3 \times \lvert\mathbb{Z}_p\rvert$	$1 \times \lvert\mathbb{G}\rvert + 4 \times \lvert\mathbb{Z}_p\rvert$	4 $\mathsf{exp}_{\mathbb{G}}$ + 3 $\mathsf{exp}_{\hat{\mathbb{G}}}$	3 $\mathsf{exp}_{\mathbb{G}}$ + 4 $\mathsf{exp}_{\hat{\mathbb{G}}}$
Our scheme	co-CDH + $\mathsf{DDH}_{\mathbb{G}}$	$n \times \lvert\mathbb{G}\rvert$	$2 \times \lvert\mathbb{Z}_p\rvert$	$1 \times \lvert\mathbb{G}\rvert + 3 \times \lvert\mathbb{Z}_p\rvert$	6 $\mathsf{exp}_{\mathbb{G}}$	6 $\mathsf{exp}_{\mathbb{G}}$

Figure: Comparison among non-interactive threshold BLS signatures

∗ Only proven secure under static corruptions in the ROM

- Our DKG decreases the broadcast channel communication by a factor 2
  Each player only sends t elements of $\mathbb{G}$ (instead of $\hat{\mathbb{G}}$)
- Like [DR24], we only need DDH in the proof of adaptive security



Summary

- Simplification of the Das-Ren threshold BLS signature which preserves
  - Adaptive security in the ROM, optimally short signatures
  - Non-interactive signing, no erasures, one-round DKG, robustness (under DDH assumption)
- Advantages:
  - Shorter private key shares (only *two* elements of $\mathbb{Z}_p$)
  - Smaller partial signatures, verification keys and communication in the DKG
- Extensions
  - Adaptively secure Naor-Pinkas-Reingold distributed PRFs (`Eurocrypt'99`)
  - Adaptively secure distributed VRFs in the ROM

Questions?

RSAC 2025 Conference

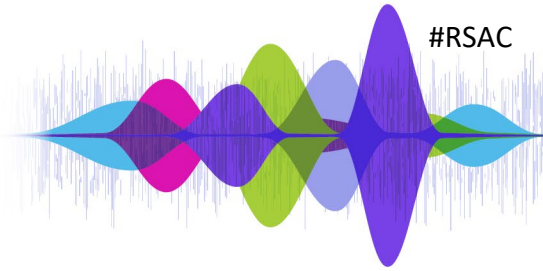
Many Voices.
One Community.

SESSION ID: CRYP-T01B

DSKE: Digital Signatures with Key Extraction

Duc V. Le

Staff Research Scientist
Visa Research
https://www.linkedin.com/in/levduc/



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Motivation: Long-lived Signature

Current Reality: Digital signatures provide authentication that remains valid indefinitely

Key Issue:

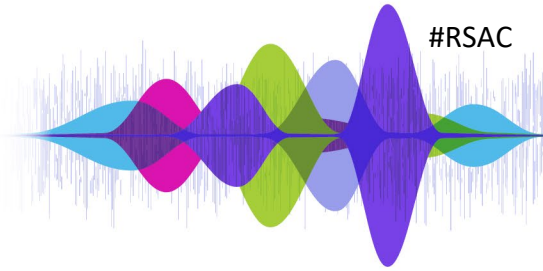
This "long-lived" property creates permanent records that can't be denied

Real-World Impact:

Legal evidence of authorship in many jurisdictions

Privacy concerns when signatures are attached to sensitive communications

Problem: The Spam Problem



- **Long-lived Signatures Enable Spam**: Current digital signature schemes have no built-in rate limiting
  - Spammers can create unlimited valid signatures with a single key
  - No effective cryptographic penalty for misuse
- **Current Anti-spam Solutions**:
  - Rely on centralized reputation systems or filtering
  - Lack cryptographic enforcement mechanisms
  - Challenge: How to limit signature creation while preserving authenticity?

Current Solutions

Off-the-Record (OTR) Communication:

Requires interactive key agreement before messaging

Not scalable for applications like email with no prior interaction

Works for small-scale, synchronous communication only

Key Rotation Approaches:

Periodic publishing of old keys (e.g., DKIM solution)

Logistical challenges in key distribution

Reliance on trusted parties or external services

Continuous maintenance overhead

Research Questions

Key Challenge:

- Can we design signatures that: Provide immediate verification
- Allow later deniability without continuous key publishing
- Penalize excessive signature creation (spam prevention)
- Function without additional infrastructure

Our Approach: DSKE - addressing both deniability and spam prevention through controlled key extraction

DSKE: Definition and Properties

A DSKE consists of 4 algorithms $(Gen, Sign, Vrfy, Extract)$:

1. A key-generation algorithm Gen takes as input a security parameter 1^n and outputs a pair of keys *(pk, sk)* :
$$Gen(1^n) \rightarrow (sk, pk)$$

2. A signing algorithm takes as input a private key *sk* and a message *m* from some message space. It outputs a signature σ :
$$Sign(sk, m) \rightarrow \sigma$$

3. A verification algorithm *Vrfy* takes as input a public key *pk*, a message *m*, and a signature σ. It outputs a bit *b.*
$$\mathrm{Vrfy}(\mathrm{pk}, \mathrm{m}, \sigma) \rightarrow \mathrm{b} \in \{0,1\}$$

4. An extraction algorithm $Extract$ takes as : The extraction algorithm is a deterministic algorithm that takes as input a set of k distinct valid message-signature pairs (m_i, σ_i) sampled from some distribution, and outputs the underlying private key sk or $\perp$.
$$Extract(\{m_i, \sigma_i\}_{i \in [k]}, pk)$$

Properties

Existential Unforgeability

Extractability under Trusted Generation (useful for deniability)

Extractability under Untrusted Generation (useful for spam prevention)

Extractability with Trusted Generation

Definition 9 (Extractability with Trusted Generation). *A digital signature scheme has a (k, δ)-extractable set when the algorithm* Extract$(\cdot)$ *on input k distinct message-signature pairs $\{(m_i, \sigma_i)\}_{i\in[k]}$ and the public key pk, such that each σ_i is a valid signature on m_i under pk, outputs the private key sk with probability δ. That is:*

$$\Pr\left[\begin{array}{l} par \leftarrow \mathsf{Setup}(1^\lambda), (pk, sk) \leftarrow \mathsf{KeyGen}(par), \\ m_i \leftarrow \mathcal{M}, \ s.t. \ m_i \neq m_j, \ for \ i, j \in [k], i \neq j, \\ \sigma_i \leftarrow \mathsf{Sign}(sk, m_i), \\ sk' \leftarrow \mathsf{Extract}\left(\{(m_i, \sigma_i)\}_{i\in[k]}, pk\right) \end{array} : pk = \mathsf{extractPK}(sk')\right] = \delta$$

Extractability with Untrusted Generation

Definition 10 (Extractability with Untrusted Generation). *A digital signature scheme has a (k, δ)-extractable set when the algorithm* Extract$(\cdot)$ *on input k distinct message-signature pairs $\{(m_i, \sigma_i)\}_{i\in[k]}$ and the public key pk generated by an adversary $\mathcal{A}$, outputs the private key sk with probability δ:*

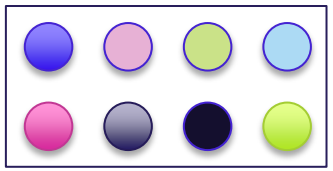
$$\Pr\left[\begin{array}{l} par \leftarrow \mathsf{Setup}(1^\lambda), (pk, \{(m_i, \sigma_i)\}_{i\in[k]}) \leftarrow \mathcal{A}(1^\lambda, par), \\ s.t.\ \mathsf{Verify}(pk, m_i, \sigma_i) = 1, \\ m_i \neq m_j,\ for\ i, j \in [k], i \neq j, \\ sk' \leftarrow \mathsf{Extract}\left(\{(m_i, \sigma_i)\}_{i\in[k]}, pk\right) \end{array} : pk = \mathsf{extractPK}(sk')\right] = \delta$$

RSAC 2025 Conference

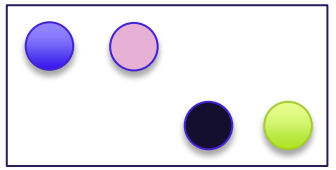
Constructions From Hash-based Signature

- **General Approach:** One-time/Few-time Hash-based signatures are those signature schemes that partially reveal the private key upon signing.

Lamport OTS

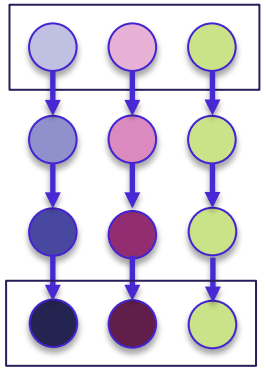


sk

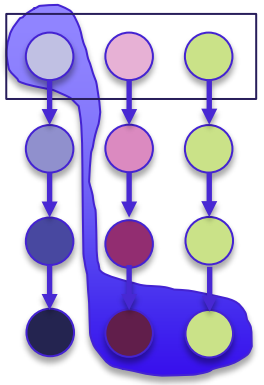


σ on 0011

Winterniz OTS



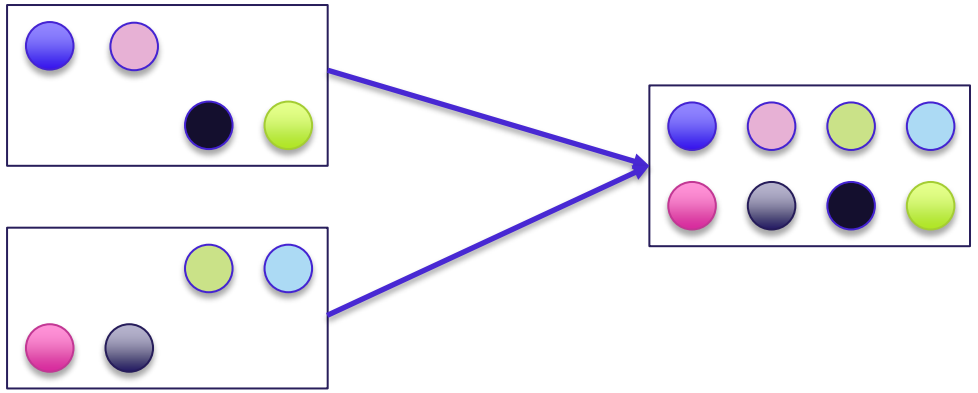
σ on 0011



RSAC 2025 Conference

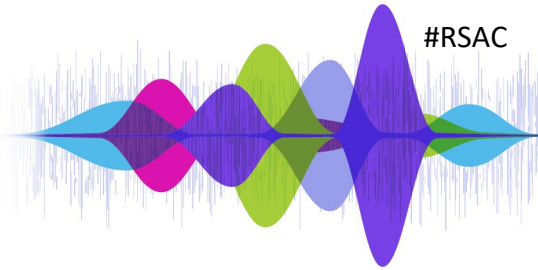
Constructions From Hash-based Signature

- Idea: With enough signatures on different message, one should be able to extract the key.



- Define Leakage function captures private key exposure from signatures
- Multiple signatures collectively provide enough information to reconstruct the full key
- Framework applies to most one-time/few time signature schemes (Biba, Winternizt) and provide the extraction probability

Limitation of Hash-based Signatures

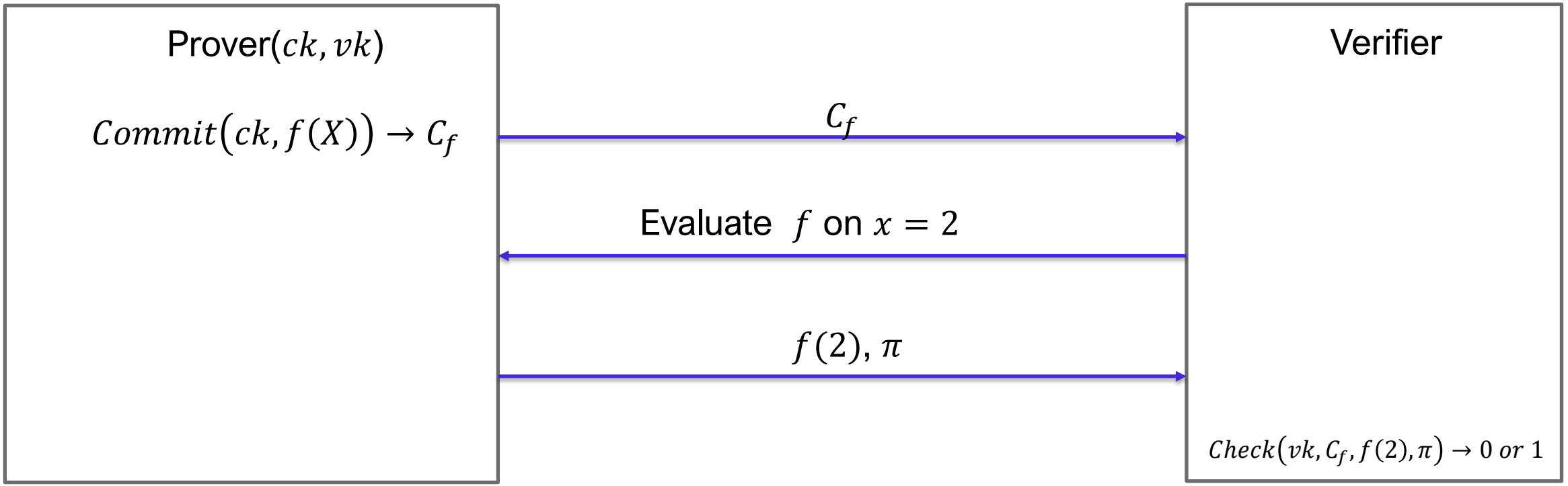


- Hash-based signatures have large signature sizes
- Do not satisfy extraction under untrusted generation

Polynomial Commitment-based DSKE

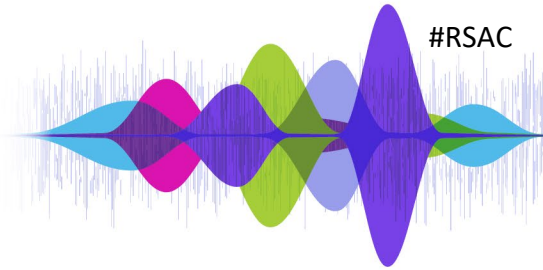
- Polynomial Commitment introduced by Kate, Zaverucha, and Goldberg

$$Setup(1^{\lambda}) \rightarrow ck, vk$$



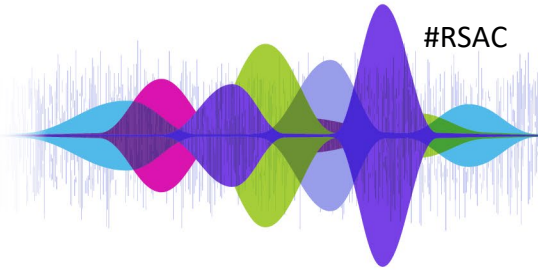
RSAC 2025 Conference

Compile Polycommit into Digital Signature



- Use Fiat-Shamir Transformation.
- Idea:
  - Private key: Polynomial $f(X)$ of degree d
  - Public key: Commitment to polynomial com_f
  - Signature: Evaluation of $f(X)$ at $H(message)$ along with the polycommit proof, π
  - Key extraction: Any $d+1$ valid signatures reveal $f(X)$ via Lagrange interpolation
- We formally proved that this construction satisfies unforgeability, extraction under both trusted and untrusted generations.

Advantages of PolyCommit Based DSKE



- **Practical Benefits**:
  - Shorter signature size
  - Deterministic extraction (100% success rate)
  - Signer can set exact threshold $(d + 1)$
  - Supports untrusted generation
  - Resists manipulation attempts

Application: GroupForge

- **Problem Addressed**: Email deniability without publishing old keys
- **GroupForge Construction**:
  - Builds on DSKE with Merkle trees
  - Time-chunk based approach like KeyForge [USENIX Sec ’21]
  - Key difference: No publishing of expired keys needed
  - Deniability emerges from extractable sets

Application: GroupForge

DSKE+Merkle Tree

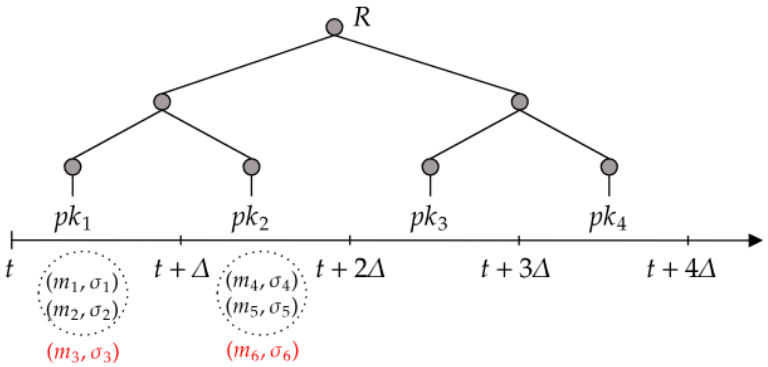


Figure 2: An example of GroupForge for $h_T = 2$ and between the interval $[t, t+2^{h_T} \cdot \Delta]$. In this example, $\{(m_i, \sigma_i)\}_{i\in[2]}$ can potentially act as the deniable group (see Section 8 for detailed discussions) for the message-signature pair (m_3, σ_3), and $\{(m_i, \sigma_i)\}_{i\in\{4,5\}}$ can act as the deniable group for the message-signature pair (m_6, σ_6).

More Efficient Construction

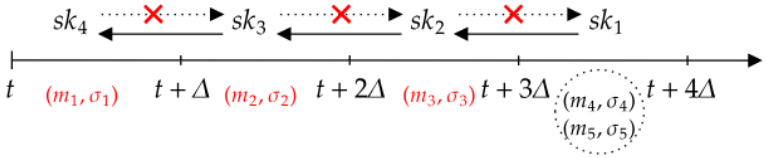


Figure 3: An example of GroupForge using FSS. The group $\{(m_i, \sigma_i)\}_{i\in\{4,5\}}$ can act as a deniable group that can reconstruct the private key to deny all message-signature pairs in the past, $\{(m_i, \sigma_i)\}_{i\in[3]}$.

GroupForge vs. Alternatives

Comparative Advantages: No reliance on continuous key publishing (unlike KeyForge)

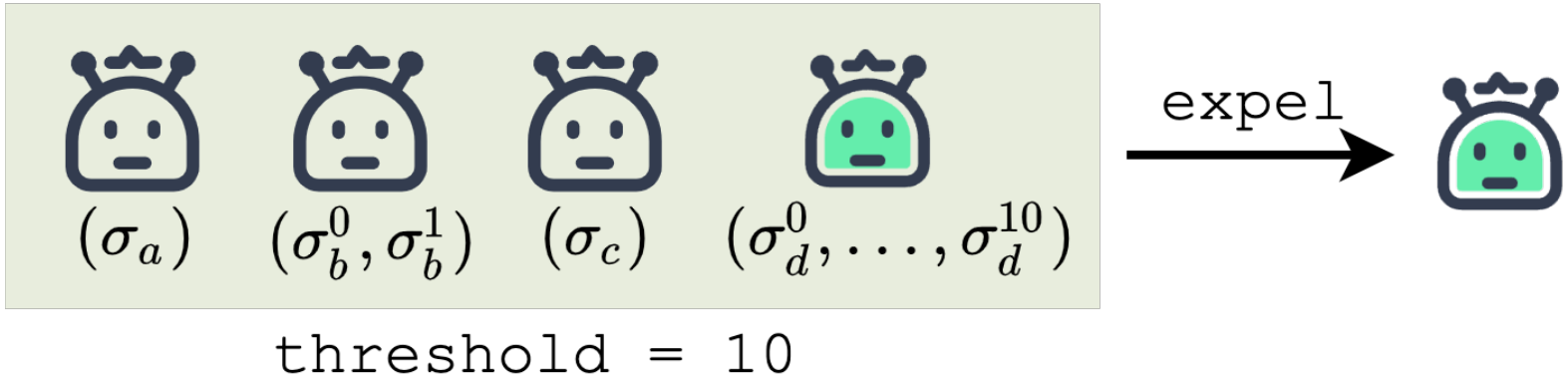
No dependence on external services (unlike TimeForge)

No need for VDFs or other expensive cryptography

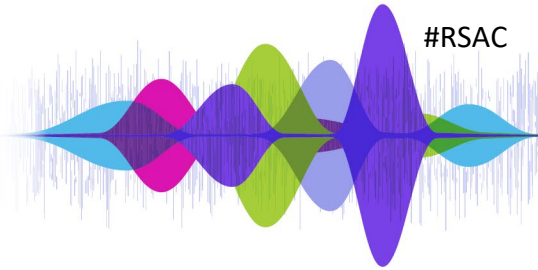
Functions as both forward-forgeable signature and short-lived signature

Application: Rate-Limiting Nullifier

- **Direct Application: Spam Prevention Mechanism**:
  - Uses key extraction as penalty for overuse
  - Private key exposure after threshold actions
  - System expels users who exceed their allocation



Anonymous RLN Extensions



- **Privacy Considerations**:
  - DSKE can be combined with NIZK to hide public keys
  - Challenge: Collecting extractable set while preserving anonymity
  - Linkable anonymity as middle ground
  - Future research direction

Implementation and Performance

Table 1: Performance of $\mathsf{DSKE}_{\mathsf{lamp}}$ and $\mathsf{DSKE}_{\mathsf{kzg}}$. Note that the probability of extracting the key in $\mathsf{DSKE}_{\mathsf{lamp}}$ is overwhelming in k, so the actual size of the extractable group can be smaller than $k = 64$.

Scheme	Group Size	Key Generation	Signing	Verification	Extraction
$\mathsf{DSKE}_{\mathsf{lamp}}$(SHA256)	$k = 64$	$451.583\mu s$	$17.625\mu s$	$160.458\mu s$	$45.083\mu s$
$\mathsf{DSKE}_{\mathsf{kzg}}$(BLS12-377)	$k = 16$	$2.795ms$	$2.425ms$	$4.949ms$	$1.785ms$
	$k = 32$	$2.825ms$	$2.523ms$	$6.123ms$	$7.813ms$
	$k = 64$	$2.852ms$	$2.756ms$	$7.202ms$	$35.341ms$
	$k = 128$	$3.624ms$	$3.404ms$	$8.045ms$	$166.433ms$

Table 2: Performance of GroupForge Constructions with $\Delta = 0.5$ hour.

Scheme	Parameters: Tree height (Duration)	Key Generation	Signing	Verification
$\mathsf{GroupForge}_{\mathsf{hash}}$ (SHA256)	$h_T = 14$ (0.93 years)	$7.407s$	$21.791\mu s$	$176.541\mu s$
	$h_T = 15$ (1.87 years)	$14.815s$	$34.708\mu s$	$177.416\mu s$
	$h_T = 16$ (3.74 years)	$29.631s$	$41.291\mu s$	$178.083\mu s$
	$h_T = 17$ (7.48 years)	$59.262s$	$45.916\mu s$	$179.374\mu s$
$\mathsf{GroupForge}_{\mathsf{kzg}}$ (SHA256, BLS12-377, $k = 32$)	$h_T = 14$ (0.93 years)	$46.294s$	$1.554ms$	$6.139ms$
	$h_T = 15$ (1.87 years)	$92.588s$	$1.591ms$	$6.140ms$
	$h_T = 16$ (3.74 years)	$185.175s$	$1.611ms$	$6.141ms$
	$h_T = 17$ (7.48 years)	$370.350s$	$1.722ms$	$6.143ms$

Conclusion and Future Work

- **Contribution Summary:**
  - DSKE primitive reconciling verification and deniability
  - Multiple constructions with security proofs
  - Practical applications in email deniability and spam prevention
- **Future Direction:**
  - Multi-Signature Extensions
  - Addition Applications in Voting, Donations

RSAC™ 2025 Conference

Many Voices.
One Community.

Thank You

Full Paper:

https://dl.acm.org/doi/10.1007/978-3-031-88661-4_7